

HTML5



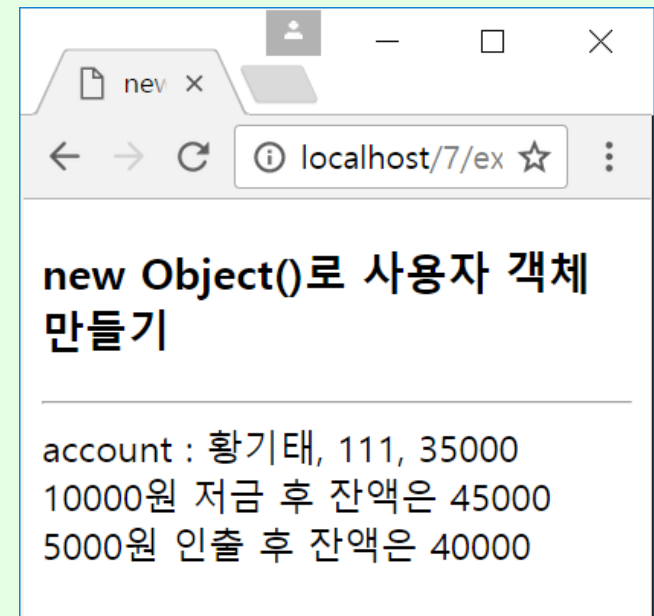
My ID (HTML5, 1st semester 2019)

성명	ID
강민수	HM01
구병준	HM02
김종민	HM03
박성철	HM04
이창호	HM05
김동영	HM06
김종건	HM07
박승훈	HM08
손성빈	HM09
손지석	HM10
안예찬	HM11
양성철	HM12
유종인	HM13
차재옥	HM14
강현이	HM15
김경미	HM16

김경영	HM17
김규년	HM18
김수정	HM19
김영록	HM20
김예원	HM21
김유빈	HM22
김정인	HM23
김주호	HM24
송다은	HM25
오주현	HM26
이한글	HM27
장세진	HM28
정주은	HM29
정호기	HM30
황혁준	HM31
김태은	HM32
이재면	HM33

Weekly plan (HTML5, 1st semester 2019)

- **wk01 : Introduction to curriculum, current state of HTML5 & github**
- **wk02 : Making HTML5 documents**
- **wk03 : iFrame, Media, my 1st Homepage & Intro to Semantic tags**
- **wk04 : Semantic tags & Web forms**
- **wk05 : CSS3 I. Basic & Box model**
- **wk06 : CSS3 II. Advanced & Animation**
- **wk07 : my 2nd Homepage**
- **wk08 : Mid-term Exam.**
- **wk09 : JS I. Data types & operators**
- **wk10 : JS II. if, loop & functions**
- **wk11 : JS III. Core objects & array**
- **wk12 : JS IV. DOM & events**
- **wk13 : JS V. events & BOM**
- **wk14 : my final Homepage**
- **wk15 : Final exam.**



예제 섭씨-화씨 온도 변환앱 (DIY-1)

4

섭씨-화씨 온도 변환
함수를 만들어서
결과를 출력하시오.

```
function c2f(c) {
```

```
}
```

Javascript :

F <-> C Temperture Converter

$$F = (9/5)*C + 32$$

섭씨온도(C)	화씨온도(F)
0	32
5	41
10	50
15	59
20	68
25	77
30	86
35	95
40	104
45	113
50	122

Save [hmx_x_temperature.html](#)

예제 섭씨-화씨 온도 변환앱 (DIY-1)

5

```
<script>
  function c2f(c) {
    var f = eval((c * 9.0) / 5 + 32);
    return f;
  }
</script>
```

```
<h3>온도 변환</h3>
<h3>F = (9/5)*C + 32</h3>
<hr />
<table border="3">
  <tr>
    <td>섭씨 온도</td>
    <td>화씨 온도</td>
  </tr>
  <script>
    for (celsius = 0; celsius <= 50; celsius = celsius + 5) {
      document.write(
        "<tr><td>" + celsius + "</td><td>" + c2f(celsius) + "</td></tr>"
      );
    }
  </script>
</table>
```

JS-project 2: JS fancy calculator

6

간단한 계산기

첫번째 정수:

두번째 정수:

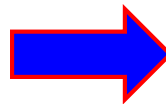
계산 결과:



Javascript : Calculator

display

7	8	9	/
4	5	6	*
1	2	3	-
0	+		
Clear	Enter		



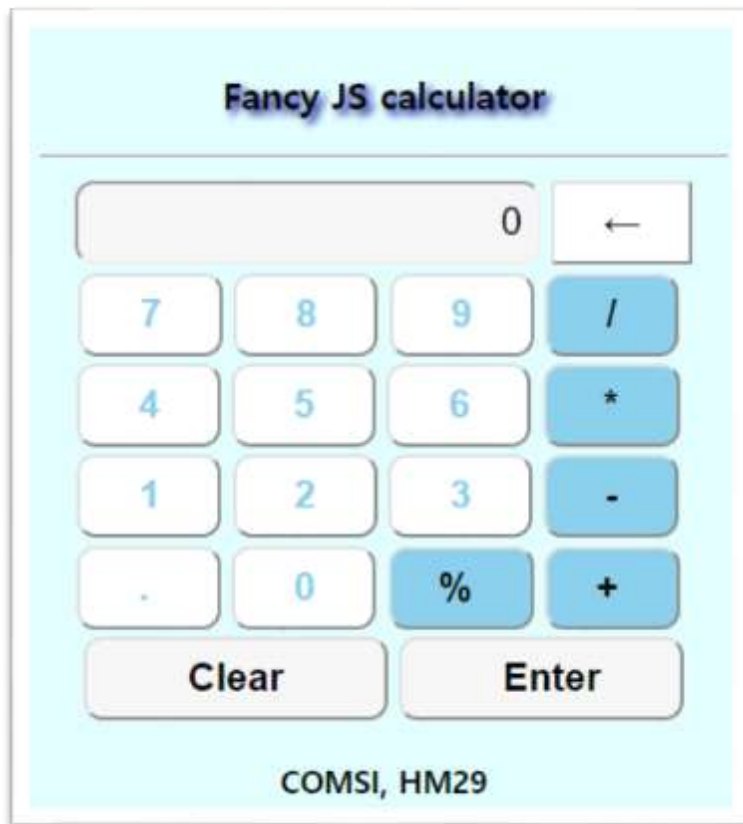
JS Calculator

<input type="text" value="0"/>			<input type="button" value="←"/>
7	8	9	/
4	5	6	*
1	2	3	-
0		.	+
Clear		Enter	

COMSL, HM00

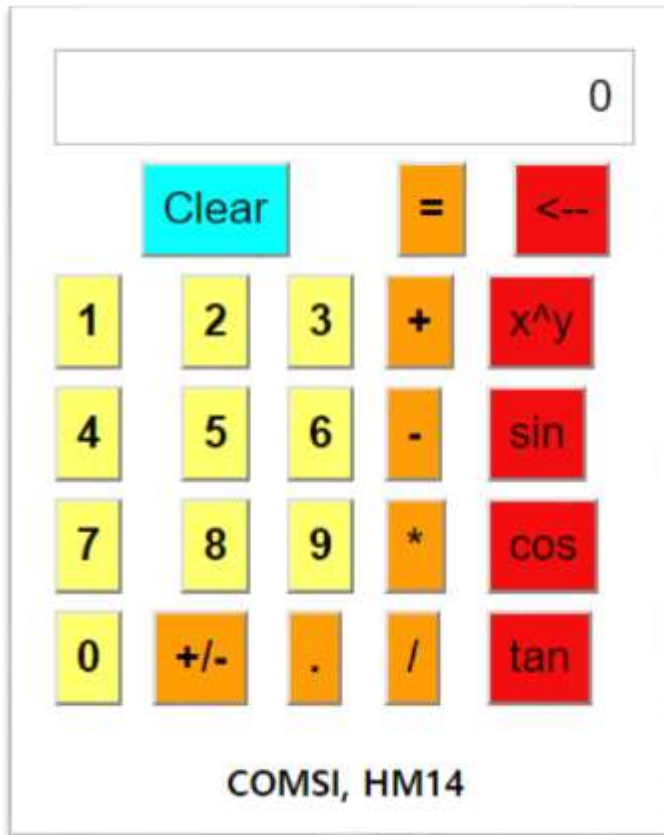
JS-project 2: JS fancy calculator

7

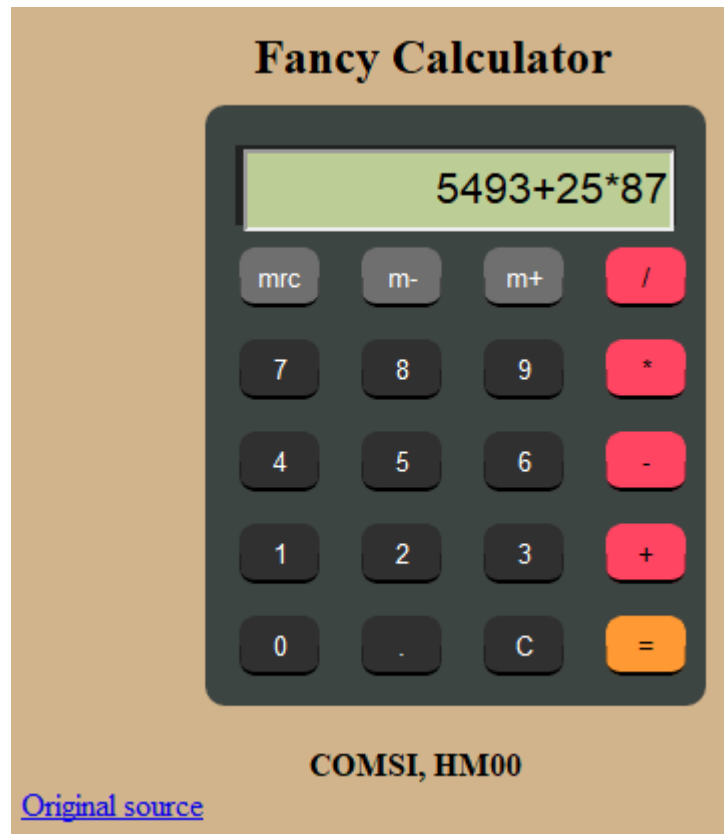


JS-project 2: JS fancy calculator

8

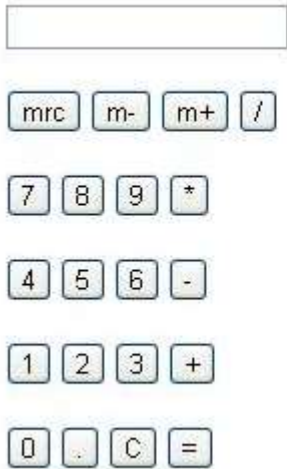


Fancy calculator



<http://www.codeproject.com/Tips/631525/Creating-Calculator-using-HTML-CSS-and-JavaScript>

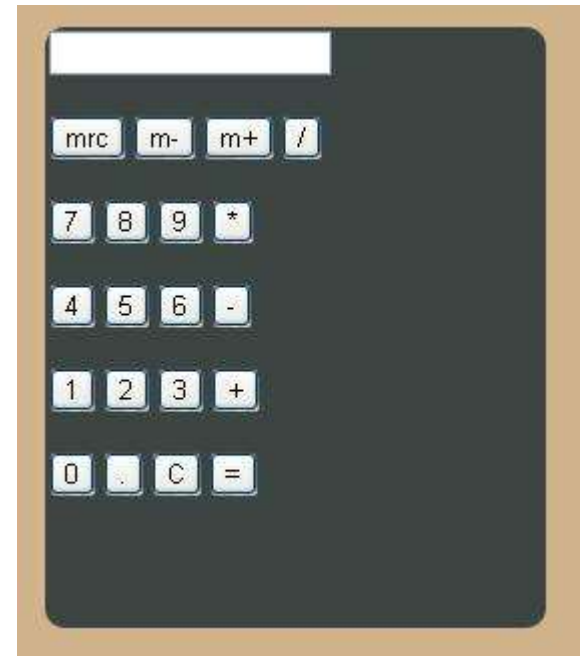
Fancy calculator – css 1



Now we have to adjust
button
width,color,background-color
or etc.

Next step is Creating
Stylesheet

```
body
{
  background-color:tan;
}
.box
{
  background-color:#3d4543;
  height:300px;
  width:250px;
  border-radius:10px;
  position:relative;
  top:80px;
  left:40%;
}
```



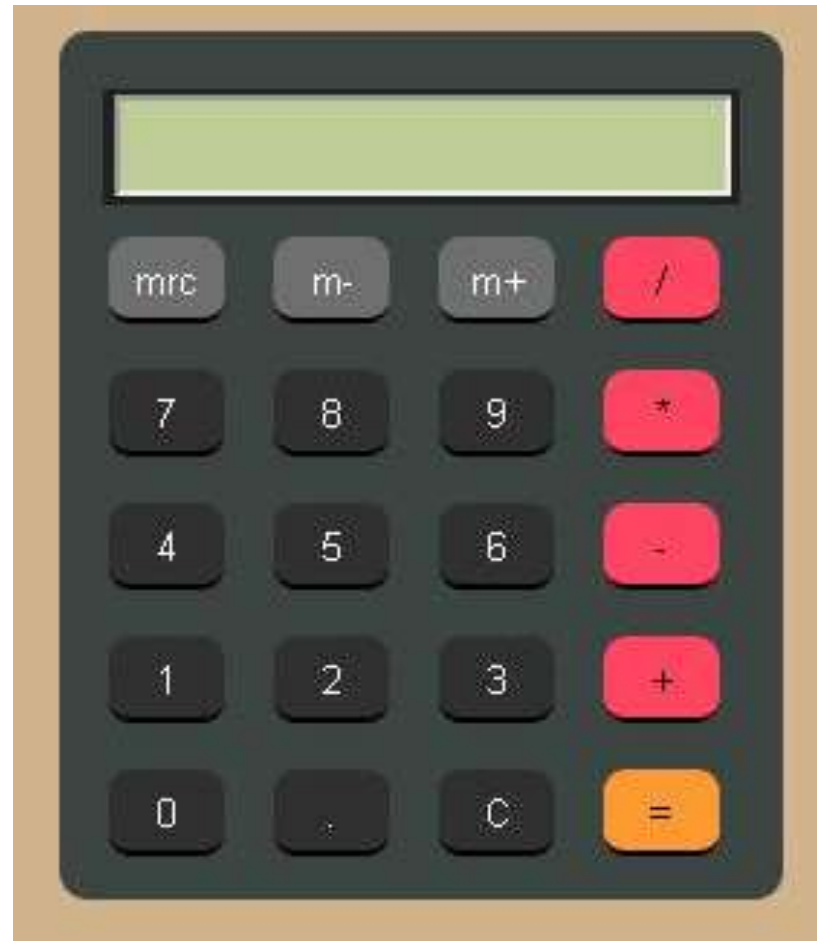
Fancy calculator – css 2



Here I adjust width of text box acc. to display

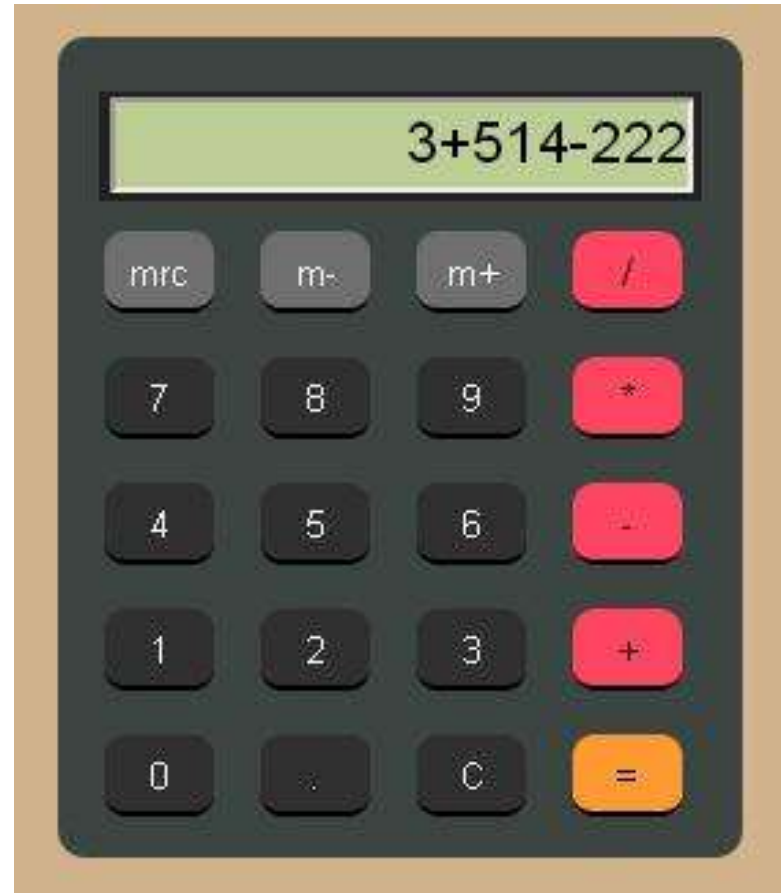
Fancy calculator – css 3

```
.keys
{
position:relative;
top:15px;
}
.button
{
width:40px;
height:30px;
border:none;
border-radius:8px;
margin-left:17px;
cursor:pointer;
border-top:2px solid transparent
}
.button.gray
{
color:white;
background-color:#6f6f6f;
border-bottom:black 2px solid;
border-top:2px #6f6f6f solid;
}
.button.pink
{
color:black;
background-color:#ff4561;
border-bottom:black 2px solid;
}
.button.black
{
color:white;
background-color:303030;
border-bottom:black 2px solid;
border-top:2px 303030 solid;
}
.button.orange
{
color:black;
background-color:FF9933;
border-bottom:black 2px solid;
border-top:2px FF9933 solid;
}
.gray:active
{
border-top:black 2px solid;
border-bottom:2px #6f6f6f solid;
}
.pink:active
{
border-top:black 2px solid;
border-bottom:#ff4561 2px solid;
}
.black:active
{
border-top:black 2px solid;
border-bottom:#303030 2px solid;
}
.orange:active
{
border-top:black 2px solid;
border-bottom:FF9933 2px solid;
}
p
{
line-height:10px;
}
```



Fancy calculator – javascript

```
function c(val)
{
    document.getElementById("d").value=val;
}
function v(val)
{
    document.getElementById("d").value+=val;
}
function e()
{
    try
    {
        c(eval(document.getElementById("d").value))
    }
    catch(e)
    {
        c('Error')
    }
}
```





07

자바스크립트 코어 객체와 배열

강의 목표

1. 객체의 기본 개념을 간단히 이해한다.
2. 브라우저가 제공하는 기본 객체(코어 객체)들의 종류를 알고 사용할 수 있다.
3. **Date** 객체를 활용할 수 있다.
4. **String** 객체를 활용할 수 있다.
5. 자바스크립트 **배열**을 만들 수 있다.
6. **Array** 객체를 이용하여 배열을 만들고 활용할 수 있다.
7. **Math** 객체를 활용할 수 있다.
8. 사용자 객체의 설계와 활용 (DIY)

객체 (object) 개념

16

- 현실 세계는 객체들의 집합
 - ▣ 사람, 책상, 자동차, TV 등
 - ▣ 객체는 자신만의 고유한 구성 속성과 방법
 - 자동차 : <색상:오렌지, 배기량:3000CC, 제조사:한성, 번호:서울1-1>
 - 사람 : <이름:이재문, 나이:20, 성별:남, 주소:서울>
 - 은행 계좌 : <소유자:황기태, 계좌번호:111, 잔액:35000원>



자동차 객체(car)

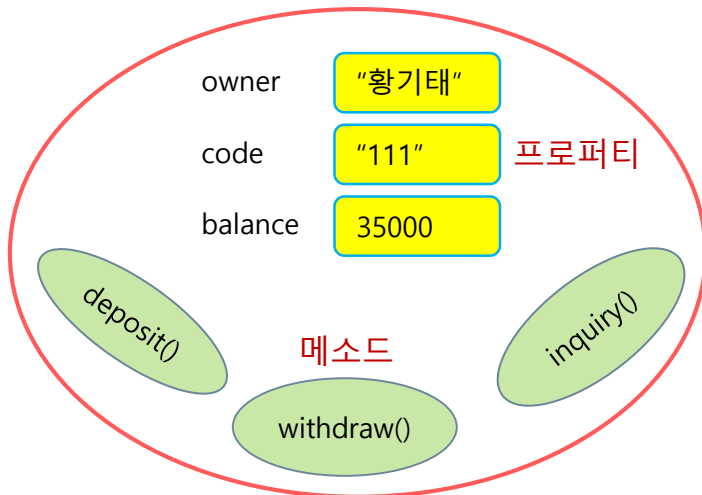


은행 계좌(account)

자바스크립트 객체

17

- 자바스크립트 객체 구성 → {...}
 - ▣ 여러 개의 **프로퍼티(property)**와 **메소드**로 구성
 - **프로퍼티** : 객체의 고유한 속성
 - **메소드**(method) : 함수



자바스크립트 객체 account

```
var account = {  
  owner    : "황기태",  
  code     : "111",  
  balance  : 35000,  
  deposit  : function() { ... },  
  withdraw : function() { ... },  
  inquiry  : function() { ... }  
};
```

account 객체를 만드는 자바스크립트 코드

json file format

자바스크립트 객체 종류

18

□ 자바스크립트는 객체 기반 언어

- ▣ 자바스크립트는 객체 지향 언어 아님 (참고, Java는 OOP)

□ 자바스크립트 객체의 유형

1. 코어 객체

- 자바스크립트 언어가 실행되는 어디서나 사용 가능한 기본 객체
- 기본 객체로 표준 객체
- Array, Date, String, Number, Math 타입 등
- 웹 페이지 자바스크립트 코드에서 혹은 서버에서 사용 가능

2. HTML DOM 객체

- HTML 문서에 작성된 각 HTML 태그들을 객체화한 것들
- HTML 문서의 내용과 모양을 제어하기 위한 목적
- DOM (Document Object Model)에 따르는 객체들
- W3C의 표준 객체

3. 브라우저 객체 (Browser object)

- 자바스크립트로 브라우저를 제어하기 위해 제공되는 객체
- BOM (Browser Object Model)에 따르는 객체들
- 비표준 객체

자바스크립트 객체 종류

19

JavaScript

- JS Array
- JS Boolean
- JS Date
- JS Error
- JS Global
- JS JSON
- JS Math
- JS Number
- JS Operators
- JS RegExp
- JS Statements
- JS String

HTML DOM

- DOM Attribute
- DOM Console
- DOM Document
- DOM Element
- DOM Events
- DOM Event Objects
- DOM History
- DOM HTMLCollection
- DOM Location
- DOM Navigator
- DOM Screen
- DOM Style
- DOM Window

<https://www.w3schools.com/jsref/default.asp>

코어 객체

(Core object)

코어 객체 (Core object)

21

- 코어 객체 종류
 - ▣ **Array, Date, String, Number, Math** 등
- 코어 객체 생성
 - ▣ **new** 키워드 이용

```
var today = new Date();           // 시간 정보를 다루는 Date 타입의 객체 생성  
var msg = new String("Hello"); // "Hello" 문자열을 담은 String 타입의 객체 생성
```

- ▣ 객체가 생성되면 객체 내부에 미리 설계된 프로퍼티와 메소드를 사용!
- 객체 접근
 - ▣ 객체와 멤버 사이에 점(.) 연산자 이용

```
obj.프로퍼티 = 값;           // 객체 obj의 프로퍼티 값 변경  
변수 = obj.프로퍼티;        // 객체 obj의 프로퍼티 값 알아내기  
obj.메소드(매개변수 값들);   // 객체 obj의 메소드 호출
```

예제 7-1 자바스크립트 객체 생성 및 활용

22

```
<!DOCTYPE html>
<html>
<head> <title>객체 생성 및 활용</title> </head>
```

```
<body>
<h3>객체 생성 및 활용</h3>
```

```
<hr>
```

```
<script>
```

```
// Date 객체 생성
```

```
var today = new Date();
```

객체 생성

```
// Date 객체의 toGMTString() 메소드 호출
```

```
document.write("현재 시간 : " + today.toGMTString()
+ "<br>");
```

메소드 호출

```
// String 객체 생성
```

```
var mystr= new String("자바스크립트 공부하기");
```

```
document.write("mystr의 내용 : " + mystr + "<br>");
```

```
document.write("mystr의 길이 : " + mystr.length + "<br>");
```

```
// mystr.length=10; // 이 문장은 오류이다.
```

프로퍼티 읽기

```
</script>
```

```
</body>
```

```
</html>
```

객체 생성 및 활용

현재 시간 : Tue, 14 May 2019 06:30:50 GMT

현재 시간 : Tue May 14 2019 15:30:50 GMT+0900 (한국 표준시)

mystr의 내용 : 자바스크립트 공부하기

mystr의 길이 : 11

자바스크립트 배열

Array

자바스크립트 배열 (Array)

24

- 배열 → **[...]**
 - 여러 개의 원소들을 연속적으로 저장
 - 전체를 하나의 단위로 다루는 데이터 구조
- 배열 생성 사례

```
var cities = ["Seoul", "New York", "Paris"];
```

cities	"Seoul"	<i>cities[0]</i>
	"New York"	<i>cities[1]</i>
	"Paris"	<i>cities[2]</i>

```
var n = [4, 5, -2, 28, 33];
```

n	4	5	-2	28	33
	<i>n[0]</i>	<i>n[1]</i>	<i>n[2]</i>	<i>n[3]</i>	<i>n[4]</i>

- 0에서 시작하는 인덱스를 이용하여 배열의 각 원소 접근**

```
var name = cities[0];  
cities[1] = "Gimhae";
```

```
// name은 "Seoul"  
// "New York" 자리에 "Gimhae" 저장
```


자바스크립트에서 배열을 만드는 방법

25

- 배열 만드는 2가지 방법
 - ▣ []로 배열 만들기
 - ▣ Array 객체로 배열 만들기
- []로 배열 만들기
 - ▣ [] 안에는 원소들의 초기 값 나열

```
var week = ["월", "화", "수", "목", "금", "토", "일"];  
var plots = [-20, -5, 0, 15, 20];
```

- ▣ 배열 크기 (dynamic sizing)
 - 배열의 크기는 고정되지 않고, 마지막 원소 추가 시 늘어남

```
plots[5] = 33; // plots 배열에 6번째 원소 추가. 배열 크기는 6이 됨  
plots[6] = 22; // plots 배열에 7번째 원소 추가. 배열 크기는 7이 됨
```

```
plots[10] = 33; // 오류. 인덱스 10은 plots 배열의 영역을 벗어났음 ??
```

예제 7-2 []로 배열 만들기

26

```
<!DOCTYPE html>
<html> <head> <title>[]로 배열 만들기</title> </head>
<body>
<h3>[]로 배열 만들기</h3>
<hr>
<script>
  var plots = [20, 5, 8, 15, 20]; // 원소 5개의 배열 생성
  document.write("var plots = [20, 5, 8, 15, 20]<br>");

  for(i=0; i<5; i++) {
    var size = plots[i]; // plots 배열의 i번째 원소
    while(size>0) {
      document.write("*");
      size--;
    }
    document.write(plots[i] + "<br>");
  }
</script>
</body>
</html>
```

[]로 배열 만들기

```
var plots = [20, 5, 8, 15, 20]
*****20
*****5
*****8
*****15
*****20
```

wk11js_ex02_array.html

예제 7-2 []로 배열 만들기 (수정)

27

```
<!DOCTYPE html>
<html> <head> <title>[]로 배열 만들기</title> </head>
<body>
<h3>[]로 배열 만들기</h3>
<hr>
<script>
  var plots = [20, 5, 8, 15, 20]; // 원소 5개의 배열 생성
  document.write("var plots = [20, 5, 8, 15, 20]<br>");
  document.write(plots + "<br><br>");
  plots[8]=99;
  document.write(plots + "<br><br>");
  document.write(plots.length + "<br><br>");

  for ( i = 0; i < plots.length; i++) {
    var size = plots[i]; // plots 배열의 i번째 원소
    while(size>0) {
      document.write("*");
      size--;
    }
    document.write(plots[i] + "<br>");
  }
</script>
</body>
</html>
```

[]로 배열 만들기

```
var plots = [20, 5, 8, 15, 20]
20,5,8,15,20
```

```
20,5,8,15,20,,,,99
```

```
9
```

```
*****20
```

```
*****5
```

```
*****8
```

```
*****15
```

```
*****20
```

```
undefined
```

```
undefined
```

```
undefined
```

```
*****
```

Array로 배열 만들기

28

□ 초기 값을 가진 배열 생성

```
var week = new Array("월", "화", "수", "목", "금", "토", "일");
```

□ 초기화되지 않은 배열 생성

▣ 일정 크기의 배열 생성 후 나중에 원소 값 저장

```
var week = new Array(7);    // 7개의 원소를 가진 배열 생성
```

```
week[0] = "월";  
week[1] = "화";  
...  
week[6] = "일";
```

□ 빈 배열 생성

▣ 원소 개수를 예상할 수 없는 경우

```
var week = new Array();    // 빈 배열 생성
```

```
week[0] = "월";           // 배열 week 크기 자동으로 1  
week[1] = "화";           // 배열 week 크기 자동으로 2
```

배열의 원소 개수, length 프로퍼티

29

□ 배열의 크기 : Array 객체의 length 프로퍼티

```
var plots = [-20, -5, 0, 15, 20];  
var week = new Array("월", "화", "수", "목", "금", "토", "일");  
var m = plots.length;    // m은 5  
var n = week.length;    // n은 7
```

□ length 프로퍼티는 읽기 전용

```
plots.length = 10; // 오류. length는 읽기 전용 프로퍼티
```

예제 7-3 Array 객체로 배열 만들기

30

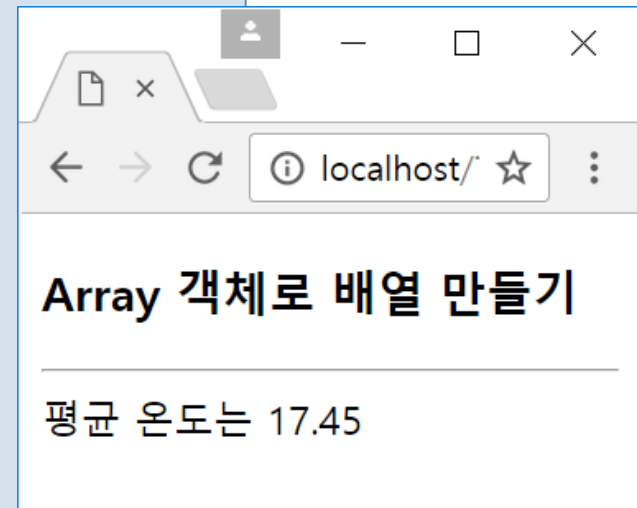
```
<!DOCTYPE html>
<html><head><title>Array 객체로 배열 만들기</title></head>
<body>
<h3>Array 객체로 배열 만들기</h3>
<hr>
<script>
  var degrees = new Array(); // 빈 배열 생성
  degrees[0] = 15.1;
  degrees[1] = 15.4;
  degrees[2] = 16.1;
  degrees[3] = 17.5;
  degrees[4] = 19.2;
  degrees[5] = 21.4;

  var sum = 0;
  for(i=0; i<degrees.length; i++)
    sum += degrees[i];

  document.write("평균 온도는 " + sum/degrees.length + "<br>");
</script>
</body>
</html>
```

배열 크기만큼 루프

배열 degrees의 크기, 6



배열의 특징

31

- 배열은 Array 객체
 - ▣ []로 생성해도 Array 객체로 다루어짐
- 배열에 여러 타입의 데이터 섞여 저장 가능

```
var any = new Array(5);    // 5개의 원소를 가진 배열 생성
any[0] = 0;
any[1] = 5.5;
any[2] = "이미지 벡터";    // 문자열 저장
any[3] = new Date();       // Date 객체 저장
any[4] = convertFunction;  // function convertFunction()의 주소 저장
```

예제 7-4 Array 객체의 메소드 활용 (Final-exam)

32

```
<!DOCTYPE html>
<html><head><title>Array 객체의 메소드 활용</title>
<script>
  function pr(msg, arr) { document.write(msg + arr.toString() + "<br>"); }
</script>
</head>
<body>
<h3>Array 객체의 메소드 활용</h3>
<hr>
<script>
  var a = new Array("황", "김", "이");
  var b = new Array("박");
  var c;

  pr("배열 a = ", a);
  pr("배열 b = ", b);
  document.write("<hr>");

  c = a.concat(b); // c는 a와 b를 연결한 새 배열
  pr("c = a.concat(b) 후 c = ", c);
  pr("c = a.concat(b) 후 a = ", a);

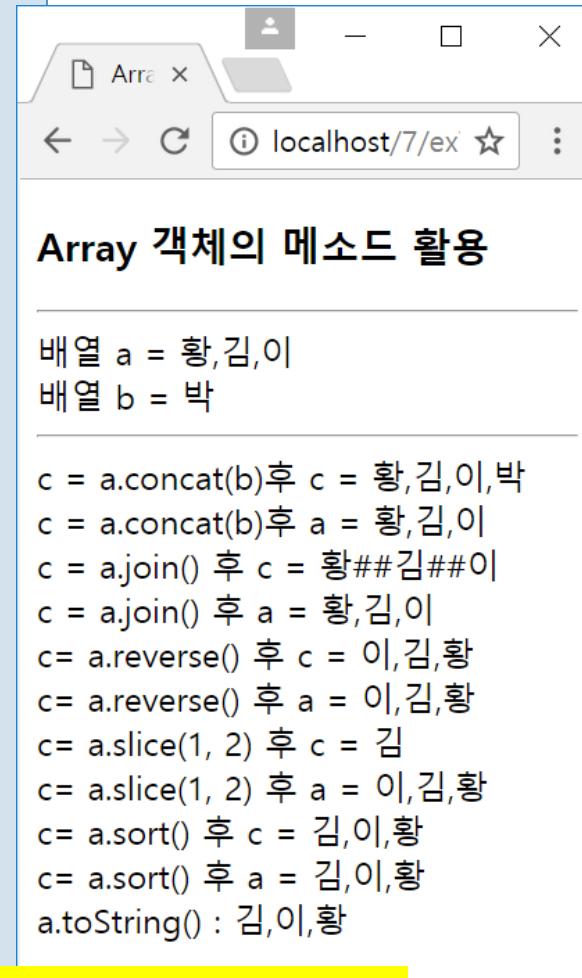
  c = a.join("##"); // c는 배열 a를 연결한 문자열
  pr("c = a.join() 후 c = ", c);
  pr("c = a.join() 후 a = ", a);

  c = a.reverse(); // a.reverse()로 a 자체 변경. c는 배열
  pr("c = a.reverse() 후 c = ", c);
  pr("c = a.reverse() 후 a = ", a);

  c = a.slice(1, 2); // c는 새 배열
  pr("c = a.slice(1, 2) 후 c = ", c);
  pr("c = a.slice(1, 2) 후 a = ", a);

  c = a.sort(); // a.sort()는 a 자체 변경. c는 배열
  pr("c = a.sort() 후 c = ", c);
  pr("c = a.sort() 후 a = ", a);

  c = a.toString(); // toString()은 원소 사이에 ","를 넣어 문자열 생성
  document.write("a.toString() : " + c); // c는 문자열
</script></body></html>
```



wk11js_ex04_array_method.html

예제 7-4 Array 객체의 메소드 활용

33

JavaScript Array slice() Method

< JavaScript Array Reference

Google

Javascript slice

Example

Select elements from an array:

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];  
var citrus = fruits.slice(1, 3);
```

The result of *citrus* will be:

Orange, Lemon

Try it Yourself »

https://www.w3schools.com/jsref/jsref_slice_array.asp

More "Try it Yourself" examples below.

Definition and Usage

The slice() method returns the selected elements in an array, as a new array object.

The slice() method selects the elements starting at the given *start* argument, and ends at, *but does not include*, the given *end* argument.

Note: The original array will not be changed.

Date 객체

Date 객체

35

□ Date 객체

- 시간 정보를 담는 객체
- 현재 시간 정보

```
var now = new Date(); // 현재 날짜와 시간(시, 분, 초) 값으로 초기화된 객체 생성
```

- 학기 시작일 2019년 3월 1일의 날짜 기억

```
var startDay = new Date(2019, 2, 1); // 2018년 3월 1일(2는 3월을 뜻함)
```

□ Date 객체 활용

```
var now = new Date(); // 현재 2019년 5월 21일 오후 2시 48분이라면  
var date = now.getDate(); // 오늘 날짜. date = 21  
var hour = now.getHours(); // 지금 시간. hour = 14
```

예제 7-5 Date 객체 생성 및 활용

36

```
<!DOCTYPE html>
<html>
<head>
<title>Date 객체로 현재 시간 알아내기</title>
</head>
<body>
<h3>Date 객체로 현재 시간 알아내기</h3>
<hr>
<script>
var now = new Date(); // 현재 시간 값을 가진 Date 객체 생성
document.write("현재 시간 : " + now.toUTCString()
    + "<br><hr>");
document.write("현재 시간 : " + now + "<br><hr>");
document.write(now.getFullYear() + "년도<br>");
document.write(now.getMonth() + 1 + "월<br>");
document.write(now.getDate() + "일<br>");
document.write(now.getHours() + "시<br>");
document.write(now.getMinutes() + "분<br>");
document.write(now.getSeconds() + "초<br>");
document.write(now.getMilliseconds() + "밀리초<br><hr>");

var next = new Date(2017, 7, 15, 12, 12, 12); // 7은 8월
document.write("next.toLocaleString() : "
    + next.toLocaleString() + "<br>");
</script>
</body>
</html>
```

Date 객체로 현재 시간 알아내기

현재 시간 : Tue, 14 May 2019 07:09:37 GMT

현재 시간 : Tue May 14 2019 16:09:37 GMT+0900 (한국 표준시)

2019년도
5월
14일
16시
9분
37초
128밀리초

next.toLocaleString() : 2017. 8. 15. 오후 12:12:12

예제 7-5 Date 객체 생성 및 활용 (수정)

37

```
// helper function to get a nicely formatted date string (for IOT date string, using RegExp)
function getKorDateString() {
    var time = new Date().getTime();
    // 32400000 is (GMT+9 Korea, GimHae)
    // for your timezone just multiply +/-GMT by 3600000
    var datestr = new Date(time + 32400000).toISOString().replace(/T/, ' ').replace(/Z/, '');
    return datestr;
}
```

<script>

```
var now = new Date(); // 현재 시간 값을 가진 Date 객체 생성
var now_Kor = getKorDateString();
```

```
document.write("현재 시간 : " + now.toUTCString() + "<br><hr>");
document.write("현재 시간 : " + now.toISOString() + "<br><hr>");
document.write("현재 시간 : " + now + "<br><hr>");
document.write("현재 시간 (KOR, IoT): " + now_Kor + "<br><hr>");
```

</script>

Date 객체로 현재 시간 알아내기

현재 시간 : Tue, 14 May 2019 07:17:56 GMT

현재 시간 : 2019-05-14T07:17:56.333Z

현재 시간 : Tue May 14 2019 16:17:56 GMT+0900 (한국 표준시)

현재 시간 (KOR, IoT): 2019-05-14 16:17:56.333

wk11js_ex05A_Date_Kor.html

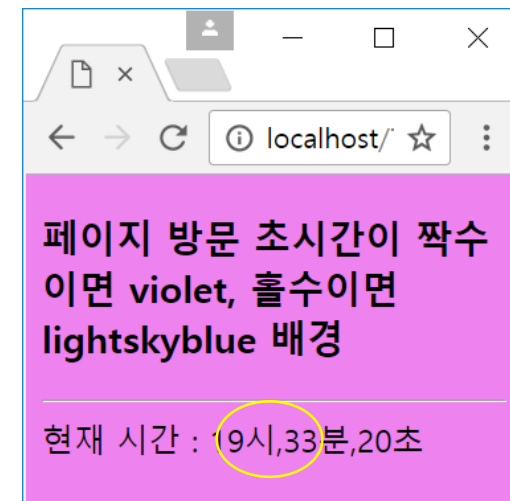
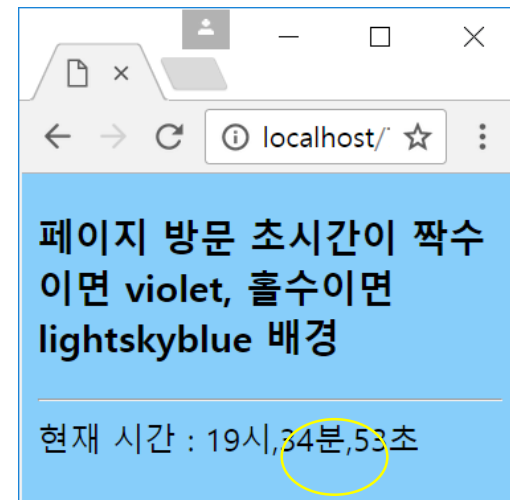
예제 7-6 방문 시간에 따라 변하는 배경색 만들기

38

```
<!DOCTYPE html>
<html>
<head>
<title>방문 시간에 따라 변하는 배경색</title>
</head>
<body>
<h3>페이지 방문 초시간이 짝수이면 violet, 홀수이면 lightskyblue 배경</h3>
<hr>
<script>
    var current = new Date(); // 현재 시간을 가진 Date 객체 생성
    if(current.getSeconds() % 2 == 0)
        document.body.style.backgroundColor = "violet";
    else
        document.body.style.backgroundColor = "lightskyblue";

    document.write("현재 시간 : ");
    document.write(current.getHours(), "시,");
    document.write(current.getMinutes(), "분,");
    document.write(current.getSeconds(), "초<br>");
</script>
</body>
</html>
```

wk11js_ex06_Date_color.html



String 객체

String 객체

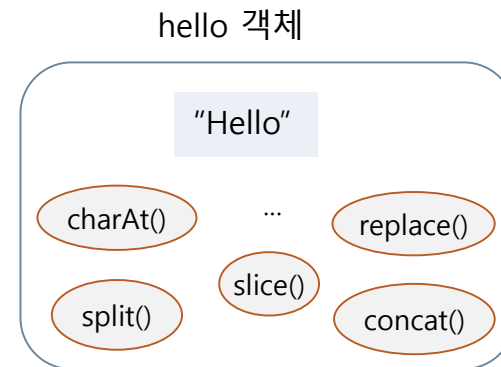
40

□ String

▣ 문자열을 담기 위한 객체

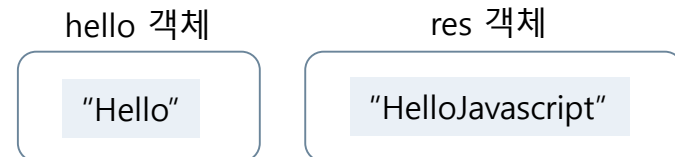
```
// 2 경우 모두 오른쪽 String 객체 생성
```

```
var hello = new String("Hello");  
var hello = "Hello";
```



▣ String 객체는 일단 생성되면 수정 불가능

```
var hello = new String("Hello");  
var res = hello.concat("Javascript");  
  
// concat() 후 hello의 문자열 변화 없음
```



String 객체의 특징

41

□ 문자열 길이

- String 객체의 length 프로퍼티 : 읽기 전용

```
var hello = new String("Hello");  
var every = "Boy and Girl";  
var m = hello.length;           // m은 5  
var n = every.length;           // n은 12
```

```
var n = "Thank you".length; // n은 9
```

□ 문자열을 배열처럼 사용

- [] 인덱스를 사용하여 각 문자 접근

```
var hello = new String(Hello");  
var c = hello[0]; // c = "H". 문자 H가 아니라 문자열 "H"
```

- Javascript에는 문자 데이터형이 없다.

예제 7-7 String 객체의 메소드 활용 (Final exam)

42

```
<!DOCTYPE html>
<html><head><title>String 객체의 메소드 활용</title></head>
<body>
<h3>String 객체의 메소드 활용</h3>
<hr>
<script>
var a = new String("Boys and Girls");
var b = "!!";
document.write("a : " + a + "<br>");
document.write("b : " + b + "<br><hr>");

document.write(a.charAt(0) + "<br>");
document.write(a.concat(b, "입니다") + "<br>");
document.write(a.indexOf("s") + "<br>");
document.write(a.indexOf("And") + "<br>");
document.write(a.slice(5, 8) + "<br>");
document.write(a.substr(5, 3) + "<br>");
document.write(a.toUpperCase() + "<br>");
document.write(a.replace("and", "or") + "<br>");
document.write("  kitae  ".trim() + "<br><hr>");

var sub = a.split(" ");
document.write("a를 빈칸으로 분리<br>");
for(var i=0; i<sub.length; i++)
    document.write("sub" + i + "=" + sub[i] + "<br>");

document.write("<hr>String 메소드를 실행 후 a와 b 변함 없음<br>");
document.write("a : " + a + "<br>");
document.write("b : " + b + "<br>");
</script>
</body>
</html>
```

wk11js_ex07_String.html



예제 7-7 String 객체의 메소드 활용 (검토!)

43

substr()과 substring()의 차이는?

```
document.write(a.substr(5, 3) + "<br>");  
document.write(a.substr(5) + "<br>");  
document.write(a.substring(5, 8) + "<br>");  
document.write(a.substring(5) + "<br>");
```

and
and Girls
and Girl
and Girls

Math 객체

Math 객체

45

□ Math

- ▣ 수학 계산을 위한 프로퍼티와 메소드 제공
- ▣ new Math()로 객체 생성하지 않고 사용 (static)

```
var sq = Math.sqrt(4);    // 4의 제곱근을 구하면 2
var area = Math.PI*2*2;    // 반지름이 2인 원의 면적
```

▣ 난수 발생

- **Math.random()** : 0~1보다 작은 랜덤한 실수 리턴
- Math.floor(m)은 m의 소수점 이하를 제거한 정수 리턴

```
// 0~99까지 랜덤한 정수를 10개 만드는 코드
for(i=0; i<10; i++) {
    var m = Math.random()*100; // m은 0~99.999... 보다 작은 실수 난수
    var n = Math.floor(m);      // m에서 소수점 이하를 제거한 정수(0~99사이)
    document.write(n + " ");
}
```

예제 7-8 Math를 이용한 구구단 연습 (DIY)

46

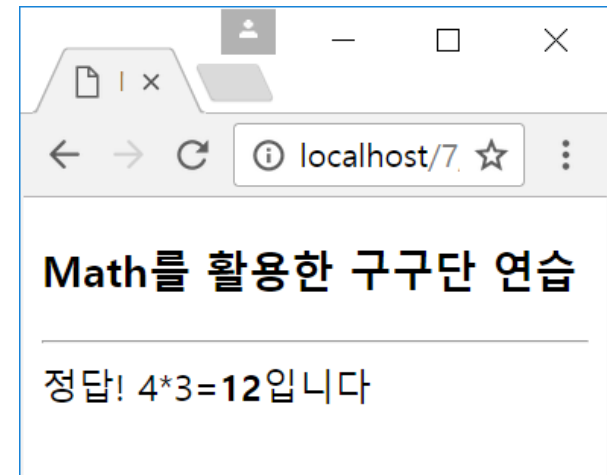
```
<!DOCTYPE html>
<html>
<head> <title>Math를 활용한 구구단 연습</title>
<script>
function randomInt() { // 1~9의 십진 난수 리턴
    return Math.floor(Math.random()*9) + 1;
}
</script>
</head>
<body>
<h3>Math를 활용한 구구단 연습</h3>
<hr>
<script>
    // 구구단 문제 생성
    var ques = randomInt() + "*" + randomInt();
    // 사용자로부터 답 입력
    var user = prompt(ques + " 값은 얼마입니까?", 0);
    if(user == null) { // 취소 버튼이 클릭된 경우
        document.write("구구단 연습을 종료합니다.");
    }
    else {
        var ans = eval(ques); // 구구단 정답 계산
        if(ans == user) // 정답과 사용자 입력 비교
            document.write("정답! ");
        else
            document.write("아니오! ");
        document.write(ques + "=" + "<strong>" + ans
            + "</strong>입니다<br>");
    }
</script>
</body>
</html>
```

wk11js_ex08_GUGUDAN.html

localhost 내용: ×

4*3 값은 얼마입니까?

확인 취소



[DIY] 정답이 맞으면 바탕이 파란색으로
틀리면 바탕이 빨간색으로 변경되게 수정하시오.

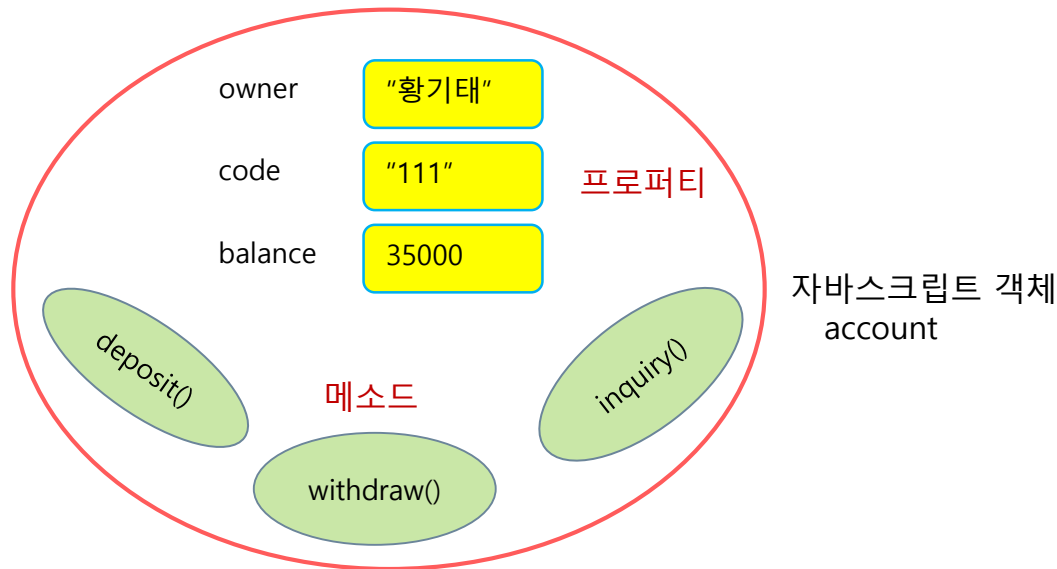
사용자 객체

User object

사용자 객체 만들기

48

- 사용자가 새로운 타입의 객체 작성 가능
- 새로운 타입의 객체 만드는 2 가지 방법만 소개
 - ▣ **new Object()** 이용
 - ▣ **리터럴 표기법** 이용
- 은행 계좌를 표현하는 account 객체



예제 7-9 new Object()로 계좌를 표현하는 account 객체 만들기

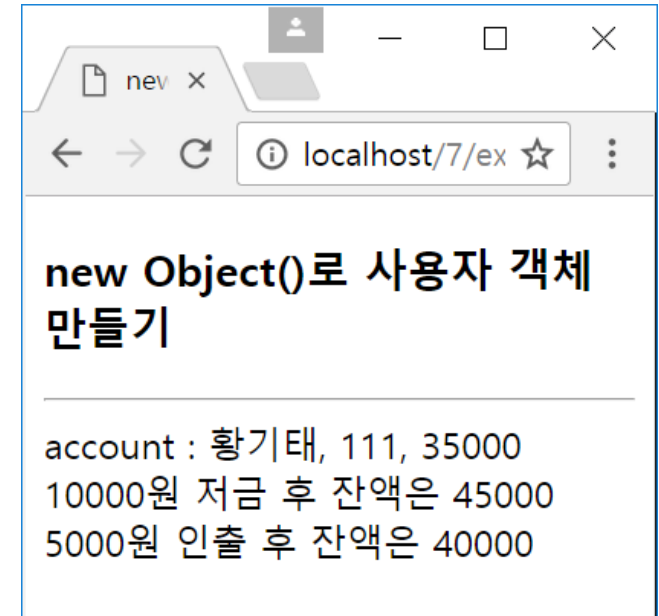
49

```
<!DOCTYPE html>
<html><head><title>new Object()로 사용자 객체 만들기</title>
<script>
  //메소드로 사용할 3 개의 함수 작성
  function inquiry() { return this.balance; } // 잔금 조회
  function deposit(money) { this.balance += money; } // money 만큼 저금
  function withdraw(money) { // 예금 인출, money는 인출하고자 하는 액수
    // money가 balance보다 작다고 가정
    this.balance -= money;
    //return money;
  }

  // 사용자 객체 만들기
  var account = new Object();
  account.owner = "황기태"; // 계좌 주인 프로퍼티 생성 및 초기화
  account.code = "111"; // 코드 프로퍼티 생성 및 초기화
  account.balance = 35000; // 잔액 프로퍼티 생성 및 초기화
  account.inquiry = inquiry; // 메소드 작성
  account.deposit = deposit; // 메소드 작성
  account.withdraw = withdraw; // 메소드 작성
</script></head>
<body>
<h3>new Object()로 사용자 객체 만들기</h3>
<hr>
<script>
  // 객체 활용
  document.write("account : ");
  document.write(account.owner + ", ");
  document.write(account.code + ", ");
  document.write(account.balance + "<br>");

  account.deposit(10000); // 10000원 저금
  document.write("10000원 저금 후 잔액은 " + account.inquiry() + "<br>");
  account.withdraw(5000); // 5000원 인출
  document.write("5000원 인출 후 잔액은 " + account.inquiry() + "<br>");
</script>
</body></html>
```

this.balance는 객체의
balance 프로퍼티



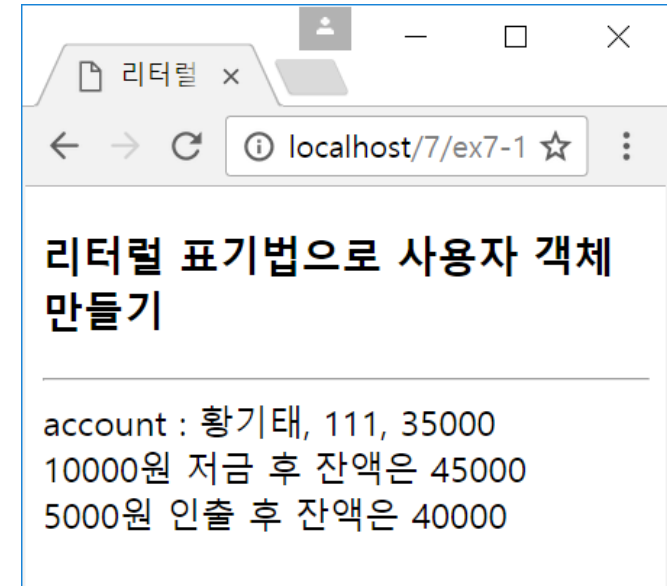
예제 7-10 리터럴 표기법(json)으로 계좌를 표현하는 account 객체 만들기

50

```
<!DOCTYPE html>
<html>
<head> <title>리터럴 표기법으로 사용자 객체 만들기</title>
<script>
//리터럴 표기법으로 사용자 객체 만들기
var account = {
  // 프로퍼티 생성 및 초기화
  owner : "황기태", // 계좌 주인
  code : "111", // 계좌 코드
  balance : 35000, // 잔액 프로퍼티

  // 메소드 작성
  inquiry : function () { return this.balance; }, // 잔금 조회
  deposit : function(money) { this.balance += money; }, // 저금. money 만큼 저금
  withdraw : function (money) { // 예금 인출, money는 인출하고자 하는 액수
    // money가 balance보다 작다고 가정
    this.balance -= money;
  }
};
</script> </head>
<body>
<h3>리터럴 표기법으로 사용자 객체 만들기</h3>
<hr>
<script>
document.write("account : ");
document.write(account.owner + ", ");
document.write(account.code + ", ");
document.write(account.balance + "<br>");

account.deposit(10000); // 10000원 저금
document.write("10000원 저금 후 잔액은 " + account.inquiry() + "<br>");
account.withdraw(5000); // 5000원 인출
document.write("5000원 인출 후 잔액은 " + account.inquiry() + "<br>");
</script>
</body> </html>
```



[DIY] 자바스크립트 객체에 관한 문제를 해결하는 두 개의 html 문서 작성.

1. [7장 연습문제-9]
리터럴 표기 방식으로 student 객체를 만들고
객체의 내용 (id, name, 전화번호, ...)을 출력하시오. (다양한 정보 추가)

파일명: `hmxx_student.html`

2. [7장 연습문제-10]
`new Object()`를 이용해서 box 객체를 만들고
(2), (3) 과정을 수행하여 결과를 출력하시오.

파일명: `hmxx_box.html`

가점: Javascript 프로그래밍 응용 능력.

Possible result ...

52

```
// 사용자 객체 만들기  
var box = new Object();  
box.color = "red";  
box.size = "10";  
box.amount = 0;  
box.fill = fill;  
box.consume = consume;
```

이 페이지 내용:

깜짝퀴즈!! 박스의 초기 amount 값을 예측해 보세요!

확인

취소

hm38 : Javascript object x

file:///E:/2017_Lec_1/HTML5/Report/

hm38 box 객체 new object 사용출력

Box : red, 10, 0

Box-Color : red
Box-Size : 10
Box-Amount : 0

document.write()로 amount 프로퍼티 값 호출2
document.write()로 amount 프로퍼티 값 호출0

깜짝퀴즈 결과!!
아닙니다! 답은0입니다.

wk11-실습 : 결과를 나의 github에 올리기

53

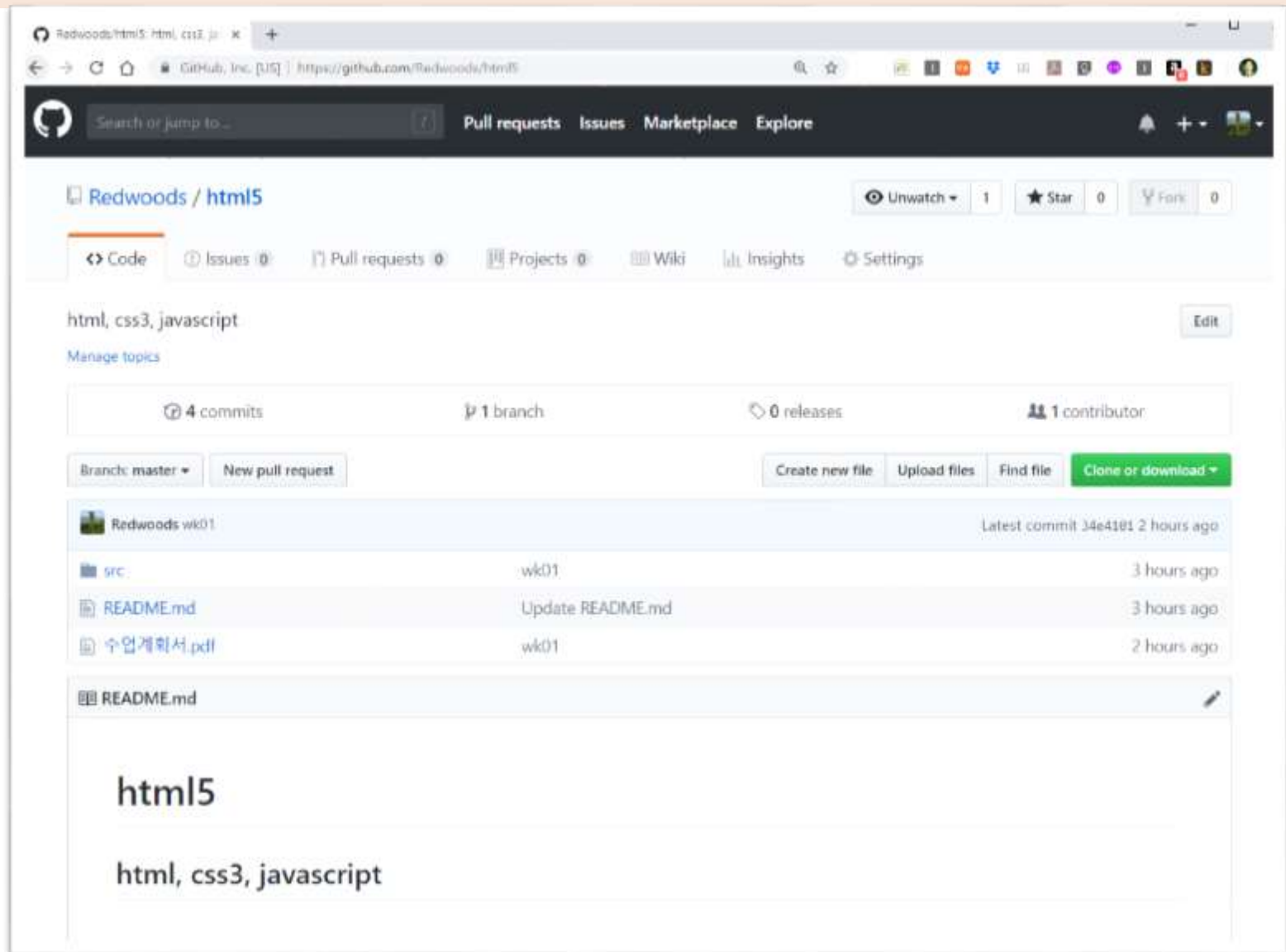
실습 결과를 github에 올립니다.

1. README.md에는 실습 결과 요약 추가 입력
2. hmxx_student.html 완성
3. hmxx_box.html 완성
4. "hmxx" repo의 wk11_src 폴더에 upload

단 업로드가 안될 경우, wk11_src.zip을 업로드
그리고 집에서 wk11_src 폴더로 다시 업로드.

HTML5 : 강의자료실

<https://github.com/redwoods/html5>



교재 WEB 강의 소개

← → ↻ ① webprogramming.co.kr ☆

명품 HTML5+ CSS3+ Javascript 웹 프로그래밍

Home Introduction Notice Board Support Code

명품 HTML5 + CSS3 + Javascript 웹 프로그래밍

HTML5로 여러분의 무한한 상상력을 표현해 보세요!

Sir Tim Berners-Lee (1955.6.8 ~)



명품 웹 프로그래밍 소개
"웹 프로그래밍을 가장 쉽게 익힐 수 있는 책"

처음 웹 프로그래밍을 공부하는 입문자들도 모든 주제를 직관적으로 이해하고 빠르게 파악할 수 있습니다.

자세히보기 →



강력한 Q&A 피드백 제공
"빠르고, 간결하고, 정확한 저자의 직접적인 답변"

'이거 이해가 잘 안되는데.. 물어볼 사람도 없고..'
더이상 고민하지 마세요.
명품 웹 프로그래밍 홈페이지에서는 누구나 저자가 직접 답변해주는 Q&A 게시판이 이용할 수 있습니다.

자세히보기 →



즉석 실행 가능한 예제 프로그램
"백문이 불여일견, 백견이 불여일타(打)!"

코드뿐만 실행되어 있는 예제들, 결과 화면이 있어도 이해가 잘 안되시죠?
예제 소스를 바탕으로, 내맘대로 수정한 코드를 즉석으로 웹 페이지로 변환해주는 예제 프로그램을 통해 모든 코드를 빠르고 쉽게 이해할 수 있습니다.

자세히보기 →



Notice

2017-01-16 15:32

Know-How

2017-01-17 14:04 관리자

55

<http://webprogramming.co.kr>

관련 WEB 강의 소개 – w3schools.com

The screenshot shows the w3schools.com website. The browser address bar displays "https://www.w3schools.com". The website has a green header with the logo "w3schools.com" and the tagline "THE WORLD'S LARGEST WEB DEVELOPER SITE". Below the header is a green navigation bar with links for "TUTORIALS", "REFERENCES", and "EXAMPLES".

On the left side, there is a sidebar menu with the following categories and links:

- HTML and CSS
 - Learn HTML
 - Learn CSS
 - Learn W3.CSS
 - Learn Colors
 - Learn Bootstrap
 - Learn Icons
 - Learn Graphics
 - Learn How To
- JavaScript
 - Learn JavaScript
 - Learn W3.JS
 - Learn JQuery
 - Learn JQueryMobile
 - Learn AppML
 - Learn AngularJS
 - Learn JSON
 - Learn AJAX
- Server Side
 - Learn SQL
 - Learn PHP
 - Learn ASP
- Web Building
 - Web Templates
 - Web Statistics
 - Web Certificates
- XML Tutorials
 - Learn XML
 - Learn XML AJAX
 - Learn XML DOM
 - Learn XML DTD
 - Learn XML Schema
 - Learn XSLT
 - Learn XPath
 - Learn XQuery

The main content area features three large sections:

- HTML**: "The language for building web pages". It includes buttons for "LEARN HTML" and "HTML REFERENCE". To the right is an "HTML Example:" box containing a code snippet for a basic HTML document structure and a "Try It Yourself" button.
- CSS**: "The language for styling web pages". It includes buttons for "LEARN CSS" and "CSS REFERENCE". To the left is a "CSS Example:" box containing a code snippet for styling a body and a heading, and a "Try It Yourself" button.
- JavaScript**: "The language for programming web pages". To the right is a "JavaScript Example:" box containing a code snippet for a JavaScript function and a "Try It Yourself" button.